Exhibit 1

IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
MIDLAND-ODESSA DIVISION

MALIKIE INNOVATIONS LTD., KEY PATENT INNOVATIONS LTD. Plaintiffs, v. MARA HOLDINGS, INC. (F/K/A MARATHON DIGITAL HOLDINGS, INC.) Defendant.	CASE NO. 7:25-CV-00222-DC-DTG JURY TRIAL DEMANDED

EXPERT DECLARATION OF PAUL D. MARTIN PH.D. IN SUPPORT OF PLAINTIFFS' RESPONSIVE CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

I. INTRODUCTION AND QUALIFICATIONS

1. My name is Paul D. Martin, Ph.D. I am the Chief Scientist at Harbor Experts, Inc. and a part-time Lecturer in the Department of Computer Science at Johns Hopkins University.

2. I have been retained by counsel for Malikie Innovations Ltd. and Key Patent Innovations Ltd. ("Plaintiffs") as an expert witness in this case. This declaration is based on my personal knowledge. If called to testify regarding the matters set forth herein, I could and would do so completely.

3. I understand that this is a patent infringement case involving the following Asserted Patents:

Asserted Patent	Asserted Claims
8,788,827	1-5
10,284,370	1-5
7,372,960	3, 6
8,666,062	1-4, 6-7
8,532,286	1, 5, 6, 9
7,372,961	1-7

4. For purposes of this declaration, I have been asked to analyze and provide my opinions about certain technical issues related to the Asserted Patents and the meaning of certain claim terms therein from the perspective of the person of ordinary skill in the art. This declaration contains my analysis and opinions, as set forth herein, based on my knowledge and experience and materials considered. In formulating my opinions in this declaration, I reviewed the materials indicated in the materials considered list below, which include the Asserted Patents and their prosecution histories.

5. This report contains my opinions formulated to date and the reasons for those opinions. I may offer further opinions based on additional review of materials in this case, including opinions and/or testimony of other expert witnesses, within the guidelines of the Court.

6. I am being compensated at a rate of $675 per hour for my work on this matter. I do not have any known financial interest in Plaintiffs or in the outcome of this litigation. My compensation in no way depends on or is affected by the results of my analysis, my conclusions or opinions, or the outcome of this litigation or any part of it.

7. A summary of my qualifications is provided below. My current curriculum vitae (CV) is attached as Appendix A to this declaration.

A. Education

8. I hold B.S., M.S.E., and Ph.D. degrees from Johns Hopkins University, with all degrees, including my doctorate, being in computer science.

B. Teaching

9. I am a part-time lecturer in the Department of Computer Science at Johns Hopkins University. I currently teach classes on computer and network security and applied cryptography, and C and C++ programming, developer tools and Linux environments. In the past I have taught classes on the topic of hardware hacking, including analyzing, modifying, and repairing computer hardware and embedded devices and vulnerability analysis and exploitation. I also occasionally substitute teach classes on network security and cryptography.

10. My classes cover a diverse array of topics including hardware and software security design, analysis and reverse engineering for vulnerability assessment; hardware and software-based attacks on computer components including RAM and CPUs; applied cryptography; computer architecture; computer networking; and component-level repair and modification. My classes also utilize hardware and software emulation and virtualization techniques in order to present students with a variety of real-world environments in which to learn and perform hands-on projects.

C. Research

11. In spring 2011, I completed my bachelor's in computer science at Johns Hopkins University. In fall 2011, I began working towards my Ph.D. in computer science, also at Johns Hopkins University, and I began researching computer security and privacy systems in order to both design novel technologies for securing computing systems and to also bridge the interface gap between users and their technology. As such, my research focused not only on developing technical solutions to novel security and privacy systems but to also presenting these solutions in a way that non-technical users could understand. Part of my research also focused on non-deterministically generating virtualized computer networks in order to provide malware education environments for students. I also developed an emulator for analyzing malware code segments as part of a student project with the NSA. This emulator was integrated into an IDA Pro plugin and was able to observe how binary instruction snippets taken from malware samples modified their computing environment.

12. During my tenure as a Ph.D. student at Johns Hopkins, I was fully funded as a research assistant and/or teaching assistant throughout all of my semesters. I was a member of the Upsilon Pi Epsilon computer science honor society, as well as its treasurer for the 2014-2015 academic year. I was also a member of the Johns Hopkins chapter of the Association for Computing Machinery. I earned an award for outstanding teaching assistant in the Computer Science department for the year 2014-2015. I co-instructed a short course called, "Introduction to Hardware Hacking," with a colleague, Dr. Michael Rushanan, which was the highest-rated course in the computer science department (based on student reviews) during the winter session in which it was offered. In this course, we offered lessons on a variety of topics including modifying game consoles and device firmware; electronics repair; binary analysis and modification, network traffic analysis; and web-based vulnerability assessment and exploitation.

13. During my time as a student at Johns Hopkins, I also participated in numerous kinds of service to the department. I represented the university as a student speaker at numerous open houses and admitted student days to help introduce prospective students to the computer science department. I was selected to speak to the external advisory board for the computer science department to provide insight into the student experience in the department. I served the department as the faculty liaison czar for several semesters, in which I attended and documented faculty meetings, which I then summarized for the students in the department. I also mentored and supervised a high school student on a year-long student project in software design and I supervised a variety of undergraduate research assistants on some of my research projects.

14. I finished my Ph.D. by successfully defending my dissertation, entitled "Securing Medical Devices and Protecting Patient Privacy in the Technological Age of Healthcare," on February 12, 2016, at the age of 26. My doctoral thesis tells the story of a secure healthcare practice of the future in which technology is seamless to use for healthcare providers while providing a much higher standard of security than is typical today. It focuses on the juxtaposition of usability and security with an emphasis on simplicity, automation, and error-proofing of security controls.

15. During my doctorate and afterward, I have produced, peer-reviewed, and published research in areas such as fingerprinting, anomaly detection, multifactor authentication, and embedded systems security. In many of these cases, I focused not only on designing novel security systems but also on designing usable web-based security systems for controlling them and understanding their data output. In fact, some of my research has formed the basis for commercial products and services. For example, in my work on integrated audit and access control, which was funded in part by Accenture Labs, I developed a Hadoop-based application to perform large-scale

statistical analysis of audit logs from an electronic medical record system. As part of this work, I also designed a web application to automatically produce human-readable reports and graphs for use in consulting. Accenture subsequently patented this technology.

16. Similarly, in my research at Applied Communication Sciences in 2013, I designed and implemented a web-based traffic visualization dashboard and analysis system for field area smart grid networks that could be used to quickly gain an understanding of the current state of a SmartGrid network as well as to detect unexpected anomalies in the network. This dashboard was able to aggregate and visualize multiple data streams in realtime. Applied Communication Sciences subsequently patented this work and continued to build on the project. To my knowledge, they sold and/or still sell this product as part of their SecureSmart Managed Security Service product offering.

17. I have worked on and published two research projects related to improving the quality and ease-of-use of authentication technologies, especially in healthcare settings. In one project, I and my co-authors designed an indoor location tracking system consisting of unspoofable Bluetooth Low Energy beacons to be used as a secondary authentication mechanism for accessing patient medical records. In another project, I and my co-authors designed an authentication bracelet that would receive a Kerberos ticket upon login to a modified computer terminal through use of low-energy electrical signals transmitted over the wearer's skin, allowing the wearer to login without a password. The bracelet was also designed to immediately lose the cryptographic secret upon being removed from the user.

18. My recent research has focused primarily on embedded systems security with an emphasis on binary analysis, anomaly detection and automated security enforcement. In one case, I designed a security system that can be soldered directly to the CPU of embedded devices in order

to perform control-flow integrity for purposes of profile building and enforcement. In another case, I designed a system to automatically discern name and version information from binaries on embedded devices in order to build a software profile of the platform configuration of the device which can then be cross-referenced with a vulnerability database. I have also sought a patent on this specific technique.

19. As of 2023 I am a member of the program committee for the IEEE Symposium on Security and Privacy. As a member of this committee, I peer review research that has been submitted for publication at the Symposium. I have recently reviewed research on memory forensics, side channel attacks, password vaults, differential privacy, computational theory, memory module design, and hardware security.

D. Patents

20. I am a named inventor on six patents. A full listing of these patents can be found on my CV, attached as Appendix A.

E. Industry

21. In 2007, I developed my first Linux distribution and an associated LiveCD build system, called PJAMAS. The Linux distribution was built on Gentoo, a source-based Linux distribution, and compiled from scratch to be a minimal desktop image that could fully boot and run on low-performance desktop computers that primarily ran the Windows operating system, and that could run Windows binaries under the WINE libraries. The ultimate goal of the distribution was to identify and remove malware from infected Windows computers. To achieve these goals, I tuned the kernel and operating system configuration, including performance-critical subsystems, to minimize RAM footprint, reduce runtime overhead, and constrain binary size, allowing the software to run on the widest possible range of Windows-era hardware. As an artifact of this

process, I also produced a general-purpose build system for creating Gentoo-based Linux LiveCDs that could be custom-designed for any purpose.

22. This early work focusing on Linux, systems, and security became the focal point of my career and led me to select security, operating systems, and embedded systems as core areas of my professional work and research.

23. I began working in the software industry in 2008, when I obtained my first independent consulting client—Brandeis University. I created a program for their computer repair shop's ticketing system to interface with a label printer. This enabled the computer repair shop to automatically print labels associated with repair tickets.

24. Shortly after finishing my engagement with Brandeis University, I began working for the Johns Hopkins University Digital Research and Curation Center as a student programmer. There, I created a batch importer for a digital archiving system called DSpace in order to enable hundreds of digitally archived works per week to automatically be added to the repository. I also ported a web interface for statistics tracking and visualization to the same digital archiving system in order to allow library staff to more easily visualize and understand data about how the digital archiving system was being used through a common web-based interface.

25. In 2009, I began working for a security consulting company called Independent Security Evaluators in Baltimore, Maryland. I worked on a wide variety of security and privacy projects including projects to test and break the DRM scheme of a magazine distribution application for IOS and Android (at the behest of the developer of said application), projects to test implementation code for cryptographic secret splitting, projects to assist with fuzz testing for security vulnerabilities of a variety of software applications, software development projects to automate testing of antivirus and antimalware solutions, and a variety of other security-related

projects. I also designed automation systems to create and restore virtualized snapshots of Windows machines on which I tested antivirus software on actual malware samples as part of a project for Consumer Reports. I worked for ISE as an intern throughout the semesters and summers until August 2011. In 2010, I also designed a security architecture for a large-scale digital curation system meant to be a major inter-university initiative to create a successor to existing digital curation systems that could be used for decades.

26. In spring 2011, I completed my bachelor's in computer science at Johns Hopkins University, and I also retained an independent consulting client, the University of Michigan, for whom I performed a penetration test and security assessment of a cloud-based research system that they planned to deploy.

27. In February 2016, I joined Harbor Labs full-time as a research scientist. In January 2019, I was promoted to Director of Firmware Analysis. In December 2023, I was promoted to Vice President of Applied Research and Technology. In October 2024, the division of Harbor Labs that I managed became Harbor Experts. I became Chief Scientist at Harbor Experts during this time. At Harbor Experts, I manage client engagements related to all of the analysis work that Harbor Labs performs. I also lead Harbor Experts' research projects.

28. My role at Harbor Experts includes managing source code review engagements. As part of this work, I have reviewed software systems of varying sizes, often totaling in the millions or billions of lines of code. I have reviewed products in the security space, television-based set top boxes, network appliances, numerous web-based enterprise systems, email management systems, telephony products, embedded system bootloaders, social network platforms, virtualization platforms and countless other products.

29. Prior to joining Harbor Experts, I was also the technical and development lead for a firmware security analysis engine for a product developed at Harbor Labs called Firmware IQ. Firmware IQ is designed to analyze the firmware of embedded devices for security vulnerabilities in an automated fashion. In one configuration of the product, a developer uploads a firmware image to a web-based portal which uses a broker to forward the firmware to an engine for analysis. The engine unpacks the firmware, breaks it into its constituent components, and then performs more than a hundred automated security checks in order to find vulnerabilities in the firmware image. The results are then reported through JSON to a web-based presentation system which displays them in an interactive fashion.

30. After my work on Firmware IQ, I developed my own Linux distribution, which I am currently in the process of releasing. This distribution is intended to address several gaps in the current market and will provide improved stability, reliability, and performance relative to existing Linux distributions.

31. A more detailed list of my engagements can be found on my attached CV.

II. MATERIALS CONSIDERED

32. I considered the following materials in preparing this declaration.

- U.S. Patent No. 8,532,286
- U.S. Patent No. 7,732,960
- U.S. Patent No. 8,666,062
- U.S. Patent No. 10,284,370
- U.S. Patent No. 8,788,827
- U.S. Patent No 7,372,961
- Exhibit 2: Certified file history of U.S. Patent No. 8,532,286
- Exhibit 5: Certified file history of U.S. Patent No. 8,532,286
- Exhibit 6: Certified file history of U.S. Patent No. 10,284,370
- Exhibit 7: Alfred J. Menezes et al., Handbook of Applied Cryptography
- Exhibit 8: Microsoft Computer Dictionary (5th ed. 2002)
- Exhibit 9: The Facts on File Dictionary of Computer Science (Valerie Illingworth & John Daintith eds., 4th ed. 2001)

- Exhibit 10: Brian Pfaffenberger, Webster's New World Computer Dictionary (10th ed. 2003)
- Exhibit 11: Dick Pountain, The New Penguin Dictionary of Computing (2001)
- Exhibit 12: Cryptographic Engineering (Çetin Kaya Koç ed., 2009)
- Dkt. 52-2: Expert Declaration of Dr. Cetin Kaya Koc in Support of MARA's Opening Claim Construction Brief
- Dkt. 52-17: National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186 (1994)
- Dkt 52-21: Peter L. Montgomery, Modular Multiplication Without Trial Division (1985)
- Dkt 52-26: Richard P. Brent et al, *Modern Computer Arithmetic Version 0.2* (2008)

III. APPLICABLE LEGAL STANDARDS

33. While I am not an attorney, I have been informed by counsel about certain legal standards that apply when determining the meaning of patent claims (a process called claim construction). My understanding of the applicable legal principles, which is based on guidance provided to me by counsel, is set forth below. I have applied these standards in rendering the opinions included in my declaration.

34. I understand that claim terms should be given their ordinary and customary meaning within the context of the patent in which the terms are used. This is the meaning that the term would have to a person of ordinary skill in the art (POSITA) at the time of the invention in light of what the patent teaches.

35. To determine how a POSITA would understand a claim term, I am informed that one should look to sources that show what a POSITA would have understood the term to mean. Such sources include the words of the claims themselves, the remainder of the patent's specification, the prosecution history of the patent (which I understand are called "intrinsic" evidence). I understand that sources outside of the intrinsic evidence (called "extrinsic" evidence) can also be considered, for example, to understand relevant scientific principles, the meaning of technical terms (or "terms of art"), and the state of the art. I understand that certain types of

extrinsic evidence, such as general purpose and scientific dictionaries, technical literature, and references illustrating the meaning of technical terms and the state of the art, may be relevant.

36. I further understand that patentees can act their own lexicographer and define a term differently than the term's plain and ordinary meaning in the art. I am informed that the patentee's own definition should control under such circumstances.

37. I understand that a dependent claim is a claim that incorporates by reference the limitations of its independent claim and of any intervening claims. As a general guideline, the scope of a dependent claim is narrower than that of its independent claim. Accordingly, dependent claims generally should be construed to be narrower in scope than their independent claim.

38. While I am not an attorney, I have been informed that a patent is presumed valid. I have also been informed that a patent claim is indefinite if, when it is read as a whole and in light of the specification and the prosecution history, it fails to inform, with reasonable certainty, those skilled in the art about the scope of the invention. I understand that absolute precision in defining the scope of the invention is not necessary, and that the level of precision depends on the nature of the subject matter.

IV. LEVEL OF ORDINARY SKILL IN THE ART

39. I understand that a person of ordinary skill in the art (POSITA) is a hypothetical person who is presumed to have been familiar with the relevant art at the time of the invention. I understand that factors to be considered in determining the level of ordinary skill in the art may include: (A) the type of problems encountered in the art; (B) prior art solutions to those problems; (C) the rapidity with which innovations are made; (D) the sophistication of the technology; and (E) the educational level of active workers in the field. I further understand that in a given case, every factor may not be present, and one or more factors may predominate.

40. For purposes of this declaration, I have been asked by counsel to assume that the Asserted Patents are entitled to the following priority dates.

Asserted Patent	Priority Date
8,788,827	January 18, 2005
10,284,370	January 18, 2005
7,372,960	December 31, 2001
8,666,062	December 31, 2001
8,532,286	July 17, 2009
7,372,961	December 26, 2001

41. Based on the above understanding, my experience in the field, and my review of the patents, I believe that a POSITA relevant to the Asserted Patents at the time of their respective priority dates would have (1) a bachelor's degree in computer science, mathematics, applied mathematics, or a related field, and two or more years of experience in applied cryptography or computer security software engineering, or (2) an advanced degree in computer science or a related field and or more years of experience in applied cryptography or computer security software engineering.

V. TECHNICAL BACKGROUND

A. Modular Arithmetic

42. Modular arithmetic is an arithmetic operation done over a finite set of elements usually denoted by a modulus, "n". Modular reduction "wraps the number around" the modulus. For example, given any integer "i", we can express it as "i=qn+r", where "q", "n", and "r" are integers. Generally, the modular reduction reduces the integer to its remainder by subtracting "n" from "i" "q" times. We can then express the remainder "r" as "r=i mod n".

1. Modular Residue

43. The elements within the modular set of "n" are the residues of the modular set defined as any integer "a" in $0 \leq a \leq n-1$. For example, under the modulus 4, the residues are [0, 1, 2, 3]. An integer "i" that is not a residue (i.e., $i > n-1$ OR $i < 0$) can be reduced to a residue. For

example, 7 is not a residue of the set modulus 4 but can be reduced to the residue 3. Two numbers under modular arithmetic are considered congruent, denoted with "≡", if they are considered to have the same residue. For example, 7 mod 4 ≡ 11 mod 4 as they both reduce to the residue 3. A negative number can be reduced to a residue by expressing the negative number as "i=qn+r". For example, -1 mod 4 = 3 as -1=(-1) 4+ 3. We often refer to the modular set "n" as $\mathbb{Z}_n$, which is a group described below in Section V.B.

2. Modular Arithmetic

44. Addition and multiplication in modular arithmetic follow from integer arithmetic. Modular addition and multiplication can be done by performing the operation and then taking the modular of it. For example, (3 x 2) + 1 mod 4 = 6 +1 mod 4 = 7 mod 4 ≡ 3.

45. Inversion and division are not as simple in modular arithmetic as there are no fractions in modular arithmetic. However, in real numbers if we wish to divide a number by "j", we multiply it by "j^{-1}". For example, in real numbers under multiplication, dividing 10 by 5 is equivalent to multiplying 10 by 1/5. Broadly, an inversion of an element under an operation is a number in the set that under the operation maps to the identity element (e.g., 1 in multiplication and 0 in addition). To find the inversion of a number "r", we find the number (or set of numbers) that solve "$e=r \times r^{-1}$", where "e" is the identity element. In real numbers, that is easily solved as "1/r" (e.g., the inverse of 5 is 1/5). In modular multiplication, the inversion is defined by "a x b mod n = 1". For example, under the modulus 5, the inverse of 4 mod 5 is 4 mod 5, because 4 x 4 mod 5 = 16 mod 5 ≡ 1 mod 5. Note, the inverse in modular arithmetic only exists when the modulus and operand are relatively prime (i.e., coprime) of each other (i.e., they have a greatest common denominator of 1). ("Being invertible modulo m is the same as being relatively prime to m" https://kconrad.math.uconn.edu/blurbs/ugradnumthy/eulerthm.pdf at pg. 1).

B. Finite Fields

1. Groups, Rings, and Fields

46. A group is a set "G" of elements and an operation "⊕" that satisfies four properties:

- *Closure: for any* $a \in G$, $b \in G$, *the element* $a \oplus b$ *is in* G.
- *Associative law: for any* $a, b, c \in G$, $(a \oplus b) \oplus c = a \oplus (b \oplus c)$.
- *Identity: There is an identity element* 0 *in* G *for which* $a \oplus 0 = 0 \oplus a = a$ *for all* $a \in G$.
- *Inverse: For each* $a \in G$, *there is an inverse* $(-a)$ *such that* $a \oplus (-a) = 0$.

https://web.stanford.edu/~marykw/classes/CS250_W19/readings/Forney_Introduction_to_Finite_Fields.pdf, at 7.3, pg. 77.

47. "The identity is then denoted by 1 (or e) and the inverse of a by a−1" (https://web.stanford.edu/~marykw/classes/CS250_W19/readings/Forney_Introduction_to_Finite_Fields.pdf, at 7.3, pg. 77). For example, the set of all integers under addition is a group, because adding one integer to another integer results in an integer. Addition is associative (i.e., can be done in any order). There is an identity element, as adding 0 to any integer will result in the same integer. Finally, there is an inverse for every integer that by adding the two will result in 0, as all integers have a negative inverse (e.g., 5 and -5). The set of all integers under multiplication is not a group because there are no inverses for multiplication and integers (e.g., 5 x 1/5 = 1, but 1/5 is NOT an integer). A group is called abelian if the operation satisfies the commutative property for every element in the group (e.g., 2+1 = 1+2).

48. A generator is an element in the group that each element of the group can be expressed under the operation and the generator. "A finite cyclic group is a finite group G with a particular element g ∈ G, called the generator, such that each element of G can be expressed as the sum, g⊕⋯⊕g, of some number of repetitions." (https://web.stanford.edu/~marykw/classes/CS250_W19/readings/Forney_Introduction_to_Finite_Fields.pdf, at 7.3, pg. 79). For example,

under modulus 4 with the group elements of [0,1,2,3], the elements of 1 and 3 are generators for the group, as 3 = 3mod4, 2 = 3 + 3 mod4, 1=3+3+3 mod4, and 0=3+3+3+3mod4.[1] The order of a finite group is the size of the smallest cycle in a cyclic group that divides the total elements of the group (e.g., a cycle of size 2 in group of size 6), where a cycle is denoted by a generator and every element. In a modulus group under multiplication where "n" is prime the order is of size "n".

49. A Field is a set with two operations satisfying the following:

> **7.4 Fields**
>
> **Definition 7.2** *A field is a set $\mathbb{F}$ of at least two elements, with two operations $\oplus$ and $*$, for which the following axioms are satisfied:*
>
> - *The set $\mathbb{F}$ forms an abelian group (whose identity is called 0) under the operation $\oplus$.*
> - *The set $\mathbb{F}^* = \mathbb{F} - \{0\} = \{a \in \mathbb{F}, a \neq 0\}$ forms an abelian group (whose identity is called 1) under the operation $*$.*
> - *Distributive law: For all $a, b, c \in \mathbb{F}$, $(a \oplus b) * c = (a * c) \oplus (b * c)$.*

https://web.stanford.edu/~marykw/classes/CS250_W19/readings/Forney_Introduction_to_Finite_Fields.pdf, 7.4, pg. 83

50. For example, all real numbers with addition and multiplication denote a field.

51. A field can be finite. For example, "A fundamental example of a finite (Galois) field is the set F_p of mod-p remainders, where p is a given prime number"(https://web.stanford.edu/~marykw/classes/CS250_W19/readings/Forney_Introduction_to_Finite_Fields.pdf, 7.4.1, pg. 84). Thus, modulus 5 with addition and multiplication forms a field.

C. Quadratic Residue and Elliptic Curves

52. A quadratic residue is a residue that is congruent to a perfect square modulus "n". for example, the quadratic residues of modulus 5 are 1 and 4, as $1 = 1^2$ mod5 and $4=2^2$ mod5. "The

[1] For any modulus "n" group, where "n" is prime, every non-identity element is a generator.

operation of computing square roots modulo n can be performed efficiently when n is a prime, but is difficult when n is a composite integer whose prime factors are unknown." Ex. 7 § 3.5. "Unlike the case where n is a prime, the problem of deciding whether a given $a \in Z^*_n$ is a quadratic residue modulo a composite integer n, is believed to be a difficult problem." Ex. 7 § 3.5.2.

53. "An elliptic curve E is a set of points of the form (x,y) where x and y are in a field F, such as the integers modulo a prime p, commonly referred to as F_p, and x and y satisfy a non-singular cubic equation, which can take the form $y^2=x^3+ax+b$ for some a and b in F. The elliptic curve E also includes a point at infinity, indicated as O. The points of E may be defined in such a way as to form a group. The point O is the identity of the group, so that O+P=P+O=P for any point P in E. For each point P, there is another point, which we will write as -P, such that P+(-P)=P+(-P)=O. " '827 Pat., 1:56-65. "For simplicity, it is preferable to work with an elliptic curve that is cyclic." '827 Pat., 2:8-9. "In an elliptic curve cryptosystem, the analogue to exponentiation is point multiplication. Thus, it is a private key is an integer k, corresponding public key is the point kP, where P is a predefined point on the curve that is part of the system parameters. The seed point P will typically be the generator G. The key pair may be used with various cryptographic algorithms to establish common keys for encryption and to perform digital signatures." '827 Pat., 2:17-24.

D. Random Number Generation

54. "A random bit generator is a device or algorithm which outputs a sequence of statistically independent and unbiased binary digits." Ex. 7 § 5.1.[2] In the context of cryptography, there are two broad classes of random number generators (RNGs): true RNGs and deterministic or pseudorandom RNGs (PRNGs).

[2] Because converting a bit string to an integer produces a number, a random bit generator may be thought of as a random number generator.

55. True RNGs collect entropy, or randomness, from their environment based on one or more naturally occurring sources of randomness. Ex. 7 § 5.2. In contrast, PRNGs are deterministic algorithms that, using a seed value, generate numbers that "appear" to be random. Ex. 7 § 5.

56. In Cryptographic algorithms, it is common to substitute a true RNG because, although "[i]deally, secrets required in cryptographic algorithms and protocols should be generated with a (true) random bit generator. However, the generation of random bits is an inefficient procedure in most practical environments. Moreover, it may be impractical to securely store and transmit a large number of random bits if these are required in applications such as the one-time pad (§6.1.1). In such situations, the problem can be ameliorated by substituting a random bit generator with a pseudorandom bit generator." Ex. 7 § 5.2.

57. A PRNG can be cryptographically secure. "A PRBG that passes the next-bit test (possibly under some plausible but unproved mathematical assumption such as the intractability of factoring integers) is called a cryptographically secure pseudorandom bit generator (CSPRBG)." Ex. 7 § 5.8.

58. "A pseudorandom bit generator is said to pass the next-bit test if there is no polynomial-time algorithm which, on input of the first l bits of an output sequence s, can predict the (l + 1)st bit of s with probability significantly greater than 1/2 ." Ex. 7 § 5.6. Equivalently, "A pseudorandom bit generator is said to pass all polynomial-time2 statistical tests if no polynomial-time algorithm can correctly distinguish between an output sequence of the generator and a truly random sequence of the same length with probability significantly greater that ½." Ex. 7 § 5.5.[3]

[3] "Although Definition 5.5 appears to impose a more stringent security requirement on pseudorandom bit generators than Definition 5.6 does, the next result asserts that they are, in fact, equivalent." Ex. 7 § 5.6.

59. While cryptographic security can be a property of a PRNG, a CSPRNG must also be used in a proper way in order for the overall system in which it is used to be considered secure. This includes using a seed that is, itself, either generated by a True RNG or by using a CSPRNG that was seeded with sufficient entropy. It is important to note that the security properties of a CSPRNG, stated differently, mean that although the output of a CSPRNG is deterministic it is not predictable to an adversary with limited computation resources without knowledge of the underlying seed.

VI. THE ASSERTED PATENTS

A. The '286 Patent

60. The '286 patent is entitled "System and Method for Reducing the Computation and Storage Requirements for a Montgomery-Style Reduction" and relates generally to "an alternative way in which to produce a Montgomery reduction," a common use method. '286 Pat., Title; Abstract.

61. In public key cryptography, certain operations use modular arithmetic. '286 Pat., 1:20-30. "[T]o multiply two numbers modulo some [number] n, the classical approach is to first perform the multiplication and then calculate the remainder."[4] *Id*. "The calculation of the remainder is referred to as reduction," and it is considered to be a slow process. *Id*. One well known method for modular reduction is Montgomery modular reduction, often referred to as Montgomery reduction. *Id*. at 1:31-35. The '286 patent relates generally to "an alternative way in which to produce a Montgomery reduction." *Id*. Abstract.

[4] The "modulo" operation calculates the remainder of dividing one number by another called the "modulus." For example, 14 mod 12 = 2 (*i.e.*, 14 ÷ 12 = 1 remainder 2).

62. Conventional Montgomery reduction reduces a value "a" using a Montgomery radix "R" to calculate aR^{-1} mod n. *Id*. at 1:47-64. A precomputed value "μ" is used to calculate a multiplier "$m=\mu a \bmod 2^w$". A multiple of the modulus "n" (i.e., m×n) is then added to the value "a" (i.e., a+m×n is computed) to "zero" out the least significant word of "a", and the result is "shifted down" to eliminate the least significant word.[5] *Id*. at 2:47-61, 5:4-17. Adding the multiple of the modulus to "a" to eliminate its least significant word cancels it. Repeating this process effects the reduction aR^{-1} mod n." *Id*. The values for μ and n are generally stored in registers throughout this process.[6] *Id*.

63. To reduce the number of stored values and computations in accordance with the invention, the '286 Patent teaches the use of "a modified reduction value" that "can be used in place of μ and n." *Id*. at 5:28-36. A "logical shift" or a "signed version" of the reduction value may alternatively be used. *Id*. Rather than being cancelled using a multiple of the modulus, the least significant word (LSW) is replaced with a different value generated using the reduction value . *Id*. at 5:45-67; Fig. 6. Replacement using the modified reduction value obviates the need to multiply by μ and determine m. *Id*. "This … avoids both the multiplication necessary to compute m and the storage required for μ." *Id*. It also avoids the need to store n. *Id*.

1. "Montgomery style reduction"

64. I understand from counsel that MARA has proposed a construction of the preamble of claim 1 of the '286 Patent that defines "Montgomery-style reduction" as "reduction that

[5] A "shift" is a programming operation that moves the binary digits (bits) of a number to the left or right, which is equivalent to multiplying or dividing the number by a power of two. For example, 00010101 shifted to the right by two bits is 00000101 and is equivalent to dividing by 2^2 or multiplying by 2^{-2}.

[6] A register is hardware in a CPU for temporary storage of data during program execution.

proceeds by clearing the least significant portions of an unreduced operand and leaving the remainder in the more significant portions."

65. MARA's construction borrows from, but then changes, the specification's description of Montgomery reduction. The specification states that Montgomery reduction "proceeds by clearing the least-significant portions of the unreduced ***quantity***, leaving the remainder in the upper portion." '286 Patent at 1:41-46 (emphasis added). The specification also states that "[t]he implementation of Montgomery multiplication is a fundamental operation on ***values*** in Montgomery representation." *Id*. at 2:34-36. MARA's construction replaces "quantity" with "operand."

66. The patent provides an example of standard Montgomery reduction in Figure 4, which operates on "a", where "a" is represented in multiple machine words (in this case, 10 words identified as a_0 through a_9). *Id*. at 4:65-23. The patent also indicates that the process starts with an "initial representation of a" and proceeds through multiple iterations at which an intermediate (or partially reduced) representation is operated on (e.g., a', a", a"', a^{iv}, etc.). *Id*. at 4:65-5:23, Figure 4.

67. Likewise, the specification teaches that the invention (which, as claimed, uses a modified reduction value to compute a modified operand) can be performed at multiple iterations of a reduction process at which an intermediate (or partially reduced) representation is operated on (e.g., a', a", a"', a^{iv}, etc.). *See* '286 Pat., 5:45-49 (modified reduction value used "at each iteration"); 5:12-19 (obtaining and using modified reduction value "for iterations 1 to k-1"); 6:32-35 (obtaining and using modified reduction value "at each iteration"); 6:40-44 ("This process is repeated k-1 [times] …."), Figure 7. Thus, while the reduction process can start with an unreduced

operand, the invention can be performed at multiple iterations where the operand may not be unreduced.

2. "perform a replacement of a least significant word of the operand"

68. I understand from counsel that the parties dispute what it means to "perform a replacement of a least significant word of the operand." The invention of claim 1 involves "obtaining an operand" and "computing a modified operand using a reduction value … to ***perform a replacement*** of a least significant word of the operand." '286 Pat. Claim 1. The specification uses the word "replacement" to describe embodiments of the invention and does so using the ordinary sense of the word. For example, it teaches that a "modified reduction value" may be used to "***perform[] a replacement for values***." *Id*. at 3:27-39. Examples of a value replacement are provided in the specification.

69. For instance, the specification describes an example where a reduction value (n') is used to perform a replacement of the least significant word (a_0) of an operand ($a \equiv [\ldots, a_4, a_3, a_2, a_1, a_0]$) where "***the value a_0 can be replaced with*** $a_0 \times n' \times 2^w$." *Id*. at 5:59-60. That is, the least significant word of the operand is "replaced" with another value. Here, the patent thus uses the term "replaced" in the ordinary sense to describe what happens to the least significant word.

70. The patent further teaches that "using the modified reduction value n' … the least significant word ***is removed***" and a different value is added back to the remaining words. *Id*. at 6:12-19. This indicates that replacement is used in the ordinary sense of the word.

71. In my opinion, based on the intrinsic evidence above, the POSITA would have understood the '286 Patent's use of the term "replacement" to be in the ordinary sense of the word.

72. I further understand from counsel that MARA has proposed a construction of "perform a replacement of a least significant word of the operand" that requires "add[ing] a modular equivalent of the operand's least significant word." I further understand that MARA

relies on the value $a_0 \times n' \times 2^w$ in the embodiment described in the '286 Patent at 5:59-67 and 6:32-38 to support its construction.

73. The specification explains that $n' \times 2^w$ is a "shifted" version of the reduction value n'.

> To see the usefulness of this new value, it is noted that if the value n' is then ***shifted up*** by one digit, which is equivalent to multiplying by 2^w, a value is obtained that is equivalent to l mod n. Consequently, the value a_0 can be replaced with $a_0 \times n' \times 2^w$, that is, a_0 multiplied by n' ***shifted up*** one digit. To be explicit $a \equiv [\ldots, a_4, a_3, a_2, a_1, a_0]$ is replaced with $a \equiv [\ldots, a_4, a_3, a_2, a_1, 0] + a_{0 \times} n' \times 2^w$. Since $a_{0 \times} n' \times 2^w$, taken without reduction, is zero in its least significant digit (by ***the shift*** 2^w), the resulting value is 0 in its least significant digit, which is the desired low-order reduction. Typically this zero digit will be treated by shifting (either logically or physically) the value down by a digit.

74. However, claim 1 is not limited to using a shifted version of the reduction value. Claim 1 states "using a reduction value." *Id*. at Claim 1. That is consistent with the specification, which teaches that "a modified reduction value ***or*** a logical shift or signed version of such a value can be used". *Id*. at 5:31-35 (emphasis added). In light of claim 1's language and the context provided by the specification, the POSITA would understand that claim 1 is not imited to using a shifted reduction value.

75. I also understand from counsel that it may be appropriate to consider the language of other claims, including dependent claims, when trying to understand the meaning of a claim term. Here, claim 2 (which depends from claim 1) also specifies that "the reduction value is $n' = 2^{-w} \bmod n$, ***or*** a shifted ***or*** signed version of n'." *See* claim 2. The additional details about the reduction value in claim 2 are not included in claim 1. I understand from counsel that a dependent claim adds limitations to a preceding claim. Because claim 2 adds details about the reduction value that are not included in claim 1, claim 1 is broader than claim 2 and is not limited to the use of a specific type of reduction value, which is consistent with my analysis of the specification above.

3. "perform a cancellation thereof"

76. I understand from counsel that the parties dispute what it means to "perform a cancellation thereof," i.e., of the least significant word of the operand. The '286 Patent uses the term "cancellation" only in the claims plus once outside the claims (but in a similar manner as it is used in the claim). '286 Pat. at 3:31-39. Where the '286 Patent uses the term "cancellation," it does so to help describe the patent's improved technique of "using a reduction value, ***instead of a modulus used in performing a standard Montgomery reduction***, to perform a replacement of a least significant word of the operand, ***rather than perform a cancellation thereof***." *Id*. at Claim 1; *see id*. at 3:31-39 (similar).

77. As described in the patent, in standard Montgomery reduction, a multiple of the modulus "n" (i.e., m×n) is added to the operand "a"—*i.e.*, a+m×n is computed—to "zero" the least significant word of the operand, and the operand is "shifted down," thereby eliminating the least significant word. *Id*. at 2:47-53, 4:40-47, 4:65-5:23; Fig. 4. Adding the multiple of the modulus to "a" to eliminate its least significant word cancels it. Because fewer words remain in "a" after each iteration, a+m×n is computed each time, and the result is shifted, to eliminate the next least significant word. *Id*. Accordingly, cancellation is used in its ordinary sense, *i.e.*, to eliminate.

78. Note, however, the technique of the '286 Patent can also involve "zeroing." *See id*. at 5:45-49 ("A modified reduction value n' … that is used to zero the least significant non-zero word of a at each iteration, without the need to first multiply by μ an determine m is found by setting $n'=2^{-w} \bmod n$ which is also therefore in the range greater than 0 and less than n."), 6:13-19 ("the least significant word is removed from the value and is multiplied by the value n' and 2^w, and added to the remaining words of a to zero the least significant word without requiring the storage of μ or the computation and storage of the multiplier m"), 6:32-35 ("As shown in FIG. 7, by obtaining and using the modified reduction value n' and applying the relationship $a \equiv [\ldots a_4, a_3, a_2,$

a_1, 0]+a_0×n'x2^w at each iteration, the least significant word of a is zeroed and the remaining word modified."). Claim 7 even recites a step of zeroing as an additional detail (which is not recited in claim 1). *Id*., Claim 7. Accordingly, "cancellation" does not mean the same thing as zeroing.

79. I understand that Dr, Koç opines that "'cancellation,' … is a term understood in the art of modular arithmetic of performing an operation that "zeros" certain words." Dkt. 52-2 ¶51. That is, he specifically equates "cancellation" with making zeroes. I disagree with his opinion. First, as explained above, it does not make sense that "cancellation" is equivalent to zeroing in the context of the '286 patent because the '286 patent explains that preferred embodiments of the invention (and some claimed embodiments) can involve zeroing. Second, "cancellation" does not have a commonly understood meaning of zeroing in the field of art of the invention. I note that the paper in which Montgomery introduced his reduction algorithm does not use that term. Dkt. 52-21. Similarly, The Handbook of Applied Cryptography also does not use this term to describe what happens during Montgomery reduction. Further, I disagree that the single document Dr. Koç cites demonstrates that "cancelling" is a well-known term for zeroing. That document uses "cancellation" in the context of floating point subtraction, not Montgomery reduction (which is discussed elsewhere in the complete document but does not mention cancellation there despite showing the creation of zeros). In the floating-point subtraction example of that document, subtraction results in zeros, but no words are eliminated. Dkt. 52-26. So the floating-point example is not using "cancellation" in the same way as the '286 Patent.

B. The '062 and '960 Patents

80. Cryptographic protocols often require the use of "keys." '960 Pat., 1:16-55. In elliptic curve cryptography (ECC), an elliptic curve is specified with "a finite field and an equation over that finite field," where "[t]he points on the elliptic curve are the pairs of finite field elements

satisfying the equation of the curve." *Id*. To perform calculations involving points on the elliptic curve, "calculations are done in the underlying finite field." *Id*.

81. The strength of an ECC system depends on the key size, where larger keys provide more security. *Id*. at 1:66-2:7. "[H]owever, different key sizes require defining different elliptic curves over different finite fields," where, in general, the size of the finite field increases with cryptographic strength. *Id*. Accordingly, multiple finite fields may need to be supported. *Id*. at 2:8-24. Using "specific methods for each finite field leads to more efficient code since it may be optimized to take advantage of the specific finite field," but "will increase the code size dramatically." *Id*. Conversely, using "generic method[s] prevents the use of optimization techniques" which makes the code smaller but less efficient. *Id*. For example, because finite field elements are often too long to fit into one machine word (requiring programs to deal with multiple words) [7], either efficient code tailored to the number of words that must be dealt with can be used, or smaller (but slower) wordsize non-specific code can be used. *Id*. at 2:25-62.

82. To enable fast engines to be produced for specific finite fields without duplicating the bulk of instructions, *id*. at 4:10-27, the patents teach first performing a "wordsized" finite field operation on "wordsized" representations of finite field elements (producing an unreduced result), followed by a "specific" modular reduction "corresponding to the particular finite field identified" (producing a reduced result). *Id*. at 8:26-34, 8:40-60. The specific modular reduction should also "lower the length of the result to the appropriate word length of the underlying finite field." *Id*.

[7] A machine word is a unit of data (in bits) that a computer processor can handle in a single operation.

1. "finite field operation"

83. I understand that Malikie proposes construing "finite field operation" as "operation in a finite field." Malikie's construction is consistent with how a POSITA would understand the phrase in light of the specification. For example, the specification indicates that it discloses methods for "**operating** on elements **in a finite field**." '960 Pat., 4:10-12 ("In general terms, the invention provides … methods for operating on elements in a finite field."); *see id*. at Abstract (same).[8] The specification further indicates that finite field operations are operations in a finite field. *See*, *e.g.*, *id*. at 1:47-50 ("**calculations** are done **in the** underlying **finite field**"), 3:32-33 (discussing "**[i]nversion in a finite field**"), 4:4-6 ("performing **calculations in a** binary **finite field**"), 7:39-41 ("A finite field **subtraction operation** 439 is provided for use **in finite fields** F_p."), 13:43-45 ("**compute the inverse** of a value **in the finite field** $F_{2^{163}}$"). The patent's descriptions of finite field operations are consistent with the POSITA's understanding. Accordingly, Malikie's construction reflects the understanding of the POSITA.

84. I understand that MARA has proposed a construction of "operation where each operand is a finite field element." The patent, however, teaches that finite field operations can be used to operate on elliptic curve points, which are pairs of finite field elements. The specification teaches that finite field operations can operate on elliptic curve points, for example, as part of an elliptic curve operation. '960 Pat., 7:25-29 ("Each elliptic curve operation 320 requires certain finite field operations, and so accordingly pointers 330 are provided to operations in the finite field engine 400 …."); 7:48-51 ("The finite field elements [comprising an elliptic curve point] are … operated on directly by the finite field engine 400 …."), 8:5-10 (elliptic curve operations "in turn

[8] Because the '960 and '062 Patents share a common specification, I will refer to the '960 patent herein.

direct finite field operations"); 8:46-48 ("The finite field engine 400 provides finite field routines 430 for use by … the elliptic curve engine 300."). The specification also makes clear that an elliptic curve point consists of finite field elements. '960 Pat., 1:45-47 ("The points on the elliptic curve are the pairs of finite field elements satisfying the equation of the curve …."); 7:46-48 ("an elliptic curve point consists of two finite field elements"). The patent thus teaches that finite field operations can be used to operate on elliptic curve points, which are pairs of finite field elements.

85. Additionally, the specification teaches that in one embodiment "[t]he finite field elements [comprising elliptic curve points] are only operated on ***directly*** by the finite field engine." *Id*. at 7:46-52 (emphasis added). The POSITA would understand that if finite field elements are only operated on ***directly*** by the finite field engine, then they can be operated on by elliptic curve operations at least indirectly because elliptic curve operations "require," "direct," and "use" finite field operations. *See id*. at 7:25-29; 8:5-10; 8:46-48; *see also id*. at 7:46-52 ("The data passed between the engines 200 [cryptographic], 300 [elliptic curve], 400 [finite field] comprises finite field elements, since an elliptic curve point consists of two finite field elements.").

2. & 3. "reduced result" / "unreduced result"

86. I understand that Malikie proposes construing "reduced result" to mean "result of performing the claimed modular reduction," and "unreduced result" to mean "result of not performing the claimed modular reduction." Malikie's construction is consistent with the claims and specification as understood by the POSITA. The claims at issue in both patents make clear that a "reduced result" is the "result of performing the claimed modular reduction."

> **('960 patent) 3.** A method of performing a finite field operation on elements of a finite field, comprising the steps of:
>
> [a] representing each element as a predetermined number of machine words;
>
> [b] performing a non-reducing wordsized operation on said representations, said wordsized operation corresponding to said finite field operation;

[c] completing said non-reducing wordsized operation for each word of said representations to obtain an unreduced result;

[d] upon computing said unreduced result, performing a specific modular reduction of said unreduced result to reduce said unreduced result to that of a field element of said finite field to obtain a ***reduced result***; and

[e] using said ***reduced result*** in a cryptographic operation.

('062 Patent) 1. A method of performing a finite field operation on elements of a finite field, the method comprising a processor:

[a] obtaining a first set of instructions for performing the finite field operation on values representing the elements of the finite field;

[b] executing the first set of instructions to generate an unreduced result completing the finite field operation;

[c] obtaining a second set of instructions for performing a modular reduction for a specific finite field;

[d] executing the second set of instructions on the unreduced result to generate a ***reduced result***; and

[e] providing the ***reduced result*** as an output for use in a cryptographic operation.

87. The patents' specification discusses what it means to "reduce" a result through use of a specific modular reduction. For example, in the context of a finite field multiplication operation, the specification explains that after the multiplication is performed, the result of the multiplication is "passed to the finite field reduction [step]." '960 Pat., 8:26-32. A "specific reduction" is then executed that "reduces the result." *Id.*, 8:46-54; *see id*. at 4:21-24 ("unreduced value" produced and "[s]pecific reduction is then applied to the unreduced value"). Malikie's construction of "reduced result" is thus consistent with and supported by the specification.

88. As the POSITA would understand, "unreduced result" is the inverse of a reduced result, namely a "result without performing the claimed modular reduction." The POSITA would understand the claim defines a "reduced result" to be what results when the claim's "modular reduction" is performed. In context, then, "unreduced result" would be understood to be the result on which such modular reduction is not performed. And how that unreduced result is otherwise

obtained is already described in the claims. For claim 3 of the '960 patent, it is by "completing said non-reducing wordsized operation for each word of said representations to obtain an unreduced result." For claim 1 of the '062 patent, it is by "executing the [claimed] first set of instructions to generate an unreduced result."

89. The specification confirms that the POSITA would understanding that "reduced" and "unreduced" are what results with and without performing modular reductions, respectively. For example, in the context of a finite field multiplication operation, the specification explains that after the multiplication is performed, the result of the multiplication is "passed to the finite field reduction [step]." '960 Pat., 8:26-32. A "specific reduction" is then executed that "reduces the result." *Id.*, 8:46-54; *see id*. at 4:21-24 ("unreduced value" produced and "[s]pecific reduction is then applied to the unreduced value"). Malikie's construction of "unreduced result" is thus also consistent with and supported by the specification.

C. The '827 and '370 Patents

90. "Public key cryptography is based upon the generation of a key pair, one of which is private and the other public that are related by a one way mathematical function." '827 Pat., 1:24-34. In cryptosystems using the Elliptic Curve Digital Signature Algorithm, the signer selects a long term private key "d" (which is secret) and computes a long term public key "Q" that is made available to verifiers. *Id*. at 2:31-41. For any message "M," the signer can create a signature, which is a pair of integers (r, s), and any verifier can take the message M, the public key Q, and the signature (r, s) and verify whether it was created by the signer. *Id*. at 2:42-48. Typically, a signer would send their public key Q with the message, or the verifier would look it up. *Id*. at 15:15-26.

91. To avoid sending or looking up the public key, the inventors devised a technique whereby the public key can be omitted from the message and instead recovered from the signature.

Id. at 15:27-40. According to this technique, the public key is omitted from the message but recovered using the signature components (r, s) and other available information by computing $Q=r^{-1}(sR-eG)$. *Id.* Because "one can recover several candidate points Q that could potentially be the public key," the patent also teaches ways "to check that Q is [the sending] correspondent's 12 public key." '827 Pat., 15:27-61. This technique allows for savings on bandwidth and storage, which yields reduced verification times. *Id*.

1. "the electronic message omits a public key of a signer"

92. I understand the parties dispute what it means for the electronic message to "omit[] a public key of a signer." In my opinion, the POSITA would have understood this term to have its ordinary meaning, as is clear from the claim itself. Claim 1 states (in relevant part):

> **1.** A method performed by a hardware processor of a computing device, comprising:
>
> [a] receiving, by a receiver of the computing device and through a network, an electronic message including a signature, ***wherein the electronic message omits a public key of a signer***, and the signature comprises a signature on the electronic message M;
>
> [b] …
>
> [c] recovering, by the hardware processor of the computing device, the **omitted public key of the signer** based on the received first elliptic curve point and the received signature, wherein the public key comprises a second elliptic curve point in an elliptic curve group different from the first elliptic curve point, wherein the elliptic curve group includes the first and second elliptic curve points, wherein the second elliptic curve point comprises an elliptic curve point Q, wherein recovering **the omitted public key of the signer** comprises computing $Q=r^{-1}(sR-eG)$, wherein G comprises a generator of an elliptic curve group that includes the elliptic curve point R and the elliptic curve point Q, and wherein e is a hash value computed from the electronic message M; and
>
> [d] …

93. Based on the plain language of the claim, the POSITA would understand that the electronic message does not contain a signer's public key. This is confirmed by the specification and prosecution history. For example, the specification explains that f "If correspondent 12 **did**

not send her public key" with the message M, "it would be beneficial to be able to recover the public key Q from the signature." '370 Pat., 15:52-57 (emphasis added). Remarks in the prosecution history also discus "enabling recipients to recover public keys **when the public keys are omitted in a message**" Ex. 6 at 4601 (emphasis added). The only discussion of "omitting" anything from a message in the intrinsic record concerns the signer's public key and nothing else. For example, the specification consistently refers to omitting the public key, not something else. *Id*.; *see also* '370 Pat., 16:18-21 ("Omitting the **public key** from the certificate can save on bandwidth ….). The prosecution history also indicates that what's omitted is the public key and does not identify anything else that is omitted. For example, when the applicant was describing the invention, it mentioned multiple times that the "public keys" are not included in the message. Ex. 6 at 4534 ("even when the **public keys** are not included in a message"), *id*. at 4444, 4482 ("enabling recipients to determine **public keys** even when not included in a message").

94. I understand that MARA has proposed a construction of "the electronic message does not include any representation of the public key of the signer." I disagree with MARA's construction for several reasons. First, the term "representation of the public key" does not appear in the specification or prosecution history. Accordingly, the intrinsic evidence provides no help to understand what it means.

95. Second, the language "any representation of" is so vague and unbounded that it would appear to encompass anything used to calculate Q—even the signature (r, s) that the claimed method uses to recover Q—which is an absurd result. In the equation $Q=r^{-1}(sR-eG)$ in the claims, "s" and "r" are components of the signature and are used to compute Q. Omitting "r" and "s", i.e., the signature, from the message would not make sense because (a) claim 1 requires it to be included, and (b) as a practical matter, signatures are sent with the message they are securing. I understand that

MARA gave one example of what it considers to be a "representation," namely "a compressed version of the public key Q." In light of the rest of the specification, however, it is clear that if the inventors wanted the claims to encompass a compressed version of Q (which is a point on a curve), they would have known to say so. For example, the specification refers to a "compressed version of R," where R is a point on a curve. '370 at 12:15-20 (referring to "compressed version of R"). And the intrinsic evidence repeatedly and consistently refers to omitting a "public key," not a compressed version or other type of representation. Furthermore, the specification teaches that the signer ***may*** send information usable to calculate Q while omitting Q from the message. For example, the signer may send a second signature (r', s') or a "compact version of Q" (or "a more compact value derived from Q"). '370 Pat., 16:6-27. MARA's construction would appear to exclude these embodiments.

96. In my opinion, the prosecution history does not support MARA's construction. During prosecution, the applicant argued that because an alleged prior art reference included a "short term public key R" in the message, it did not disclose "omitting ***any public keys*** of the signer." Ex. 6 at 4448 (emphasis added). At the time, however, the claim required the message to omit "***any public key*** of a signer." *Id*. at 4440. (emphasis added).

> 1. (**Currently Amended**) A method, comprising:
>
> receiving, through a network, an electronic message including a signature**, wherein the electronic message omits any public key of a signer**;
>
> obtaining an elliptic curve point associated with a signature component from ~~a~~ **the** signer;
>
> generating, using data processing apparatus, a public key of the signer based on the elliptic curve point and the signature; and
>
> verifying the signature using the public key.

97. As the POSITA would understand, the applicant did not argue that the reference failed to teach omitting "any public key" merely because the short-term public key in the message could be used to compute the public key Q. Instead, the POSITA would have understood the

applicant to have been making its argument based on the fact that the short term public key was an un-omitted "public key" of the signer in contrast the claims. Ex. 6 at 4448. The Fact that the claims did not yet even refer to Q further indicates that whether something could be used to compute Q was not the basis of the argument. *See above*.

98. The applicant subsequently amended their claims again, this time replacing "***any*** public key" with "***a*** public key" that specifically corresponds to "a second elliptic curve point" comprising a point "Q." *Id*. This amendment introduced Q for the first time into the claims.

> 1. (**Currently Amended**) A method **performed by a hardware processor of a computing device**, comprising:
>
> receiving, at ~~**a verifier**~~ **the computing device** and through a network, an electronic message including a signature, wherein the electronic message omits **[[any]] a** public key of a signer**, and the signature comprises a signature on the electronic message *M***;

> ~~**generating**~~**recovering**, at the ~~**verifier and using data processing apparatus**~~ **computing device**, **[[a]] the omitted** public key of the signer based on the **received** first elliptic curve point and the **received** signature, wherein the public key comprises a second elliptic curve point in an elliptic curve group different from the first elliptic curve point, wherein the elliptic curve group includes the first and second elliptic curve points**, wherein the second elliptic curve point comprises an elliptic curve point *Q*, wherein recovering the omitted public key of the signer**

Ex. 6 at 4595. It is therefore my opinion that even if the previous claims required omitting other public keys that can be used to calculate Q, the issued claims do not.

2. "verifying that the second elliptic curve point Q represents the public key of the signer"

99. In my opinion, "verifying that the second elliptic curve point Q represents the public key of the signer" has its plain and ordinary meaning to a POSITA. Starting with the claim, claim 1 recites "generating … a public key," where "the public key … ***comprises*** a … point Q," and "generating the public key … ***comprises*** computing Q." '827 Pat., Claim 1. The fact that one comprises the other (and that generating one comprises computing the other) demonstrates that the

public key and the point Q are distinct things. Similarly, the fact that claim 2 recites a step to verify that one represents the other further demonstrates that they are distinct things. Claim 2 then requires verifying that the public key is Q:

> **2.** The method of claim 1, further comprising ***verifying that the second elliptic curve point Q represents the public key of the signer***.

100. The specification also indicates to a POSITA that verifying that the point Q represents the public key makes sense. As the specification explains, "one can recover several candidate points Q that could potentially be the public key," thus "the [receiving] correspondent 14 needs a way to check that Q is [the sending] correspondent's 12 public key." '827 Pat., 15:27-29. The patent provides examples of how that could be done. For example, a certificate from a certificate authority (CA) can be provided to allow the verifier to test if a CA signature (corresponding to the sender) verifies using Q. If so, Q was computed correctly. *Id*. at 15:40-55. Alternatively, the patent teaches that Q can be checked against "some more compact value derived from Q, such as the half of the bits of Q." *Id*. at 15:57-60.

101. I understand that MARA has proposed a construction of "verifying that the second elliptic curve point Q represents the second elliptic curve point Q." In my opinion, this construction renders claim 2 non-sensical. It does not make sense to verify that Q is Q because the public key and the point Q are distinct things. And, moreover, verifying that Q is Q is not the purpose of the verification of claim 2. As explained, multiple values of Q can be recovered, so there is a need to check whether Q actually represents the public key.

D. The '961 Patent

102. Public key cryptosystems can be used to digitally sign messages to authenticate the sender. '961 Pat., 1:26-32. The sender signs the message with their private key, and a recipient can verify the message by applying the sender's corresponding public key. *Id*. To be secure, the

private key must remain secret, so protocols have been developed that incorporate additional, short-term keys (e.g., ephemeral keys) that are used temporarily. *Id*. at 1:33-50. An ephemeral private key is usually generated by a random number generator, thus it will have a uniform distribution throughout the range of possible values. *Id*. at 1:33-50, 2:15-32. But techniques for generating random numbers (including techniques that generate a seed value from a random number generator and perform a secure hash function on it) can inadvertently introduce a bias that causes values to be selected from certain intervals more often, which can be exploited to discover private keys, rendering the system insecure. *Id*. at 2:23-61.

103. The '961 Patent teaches techniques for key generation "in which any bias is eliminated during the selection of the key." *Id*. at 3:1-3. The process includes generating a seed value from a random number generator, hashing it, determining whether the output is within an acceptable range, and accepting it if it is or rejecting it if it's not. *Id*. at 3:64-4:17. If the output is rejected, the method is repeated such that either another seed value is generated by the random number generator or the output is incremented, for example, with a deterministic function. *Id*.; *see id* at 4:18-52.

1. "random number generator"

104. I understand that Malikie proposes construing "random number generator" to mean "computer instructions capable of generating values according to a uniform random probability distribution." Malikie construction is consistent with how the POSITA would have understood "random number generator."

105. First, Malikie's construction is consistent with the patent's specification. The '961 Patent uses "random number generator" consistently with its ordinary meaning, which is not limited to true random number generators. The '961 patent explains that an RNG selects values with "a uniform distribution throughout the defined interval" of possible values. '961 Pat., 2:15-

17, 2:29-32. The patent consistently refers to "random number generator" in this ordinary sense without ever limiting it to true random number generators or excluding pseudorandom number generators. *See* '961 Pat. at 1:41-43, 2:29-32, 2:40-43, 3:33-38, 3:64-66, 4:7-10, 4:13-16, 4:18-23, 4:37-39, 4:50-52, 5:1-4. And it does so regardless of whether it is describing the generation of a key (*e.g.*, k) or a seed value (*e.g.*, SV). *Id.*

106. That is consistent with the well-known goal of an RNG. For example, the Handbook of Applied Cryptography repeatedly indicates that a random number generator (or random bit generator) generates values according to a uniform random distribution. Exhibit 7 at 170 ("A random bit generator can be used to generate (**uniformly distributed**) **random numbers**."), 40 (examples of a number "**generated randomly from a uniform distribution**" using a container of numbered balls). Technical dictionaries also confirm the common understanding that random number generators produce values are unpredictable and not more likely to occur than others. Ex. 8 at 438 (defining "random number generation" as the "[p]roduction of an unpredictable sequence of numbers in which no number is any more likely to occur … than any other").

107. Moreover, the POSITA would understand that generating random values according to a uniform distribution can be, and often is, done using pseudorandom number generators (or PRNG). For example, the Handbook of Applied Cryptography describes such a pseudorandom number generator. Ex. 7 at 186 (describing Micali-Schnorr pseudorandom bit generator that relies on values "indistinguishable … from the **uniform distribution** of integers in the interval [0, *n*-1]"). Indeed, the Handbook of Applied Cryptography, which is a well-known and commonly referenced text (which I understand Dr. Koç, MARA's expert, also relies on) makes clear that

pseudorandom number generators in the context of cryptography generate values according to a uniform distribution.

> The term ***random numbers***, when used in the context of identification and authentication protocols, **includes pseudorandom numbers** which are unpredictable to an adversary …; this differs from randomness in the traditional statistical sense. In protocol descriptions, **"choose a random number" is usually intended to mean "pick a number with uniform distribution from a specified sample space" or "select from a uniform distribution."**

Ex. 7 at 398 (emphasis added).

108. This is consistent with how RNGs would have been understood by a POSITA in the context of cryptography. The POSITA would understand that RNGs fall into two main categories: (1) "true" RNGs and (2) deterministic (pseudorandom) RNGs. "True" RNGs typically generate values using physical or natural sources of randomness. Ex. 7 at 40-41 ("most *true sources* of random sequences … come from *physical means*"), 171 ("A (true) random bit generator requires a naturally occurring source of randomness."), 172 (listing examples of natural and physical sources of randomness for hardware and software generators). Such natural sources can include, for example, time between emission of particles during radioactive decay, thermal noise, frequency instability of a free running oscillator, capacitor discharge over a fixed time, latency caused by air turbulence in a sealed disk drive, sound from a microphone or video input from a camera. Ex. 7 at 172.

109. Deterministic RNGs (pseudorandom number generators) typically generate values using deterministic algorithms (algorithms that produce the same output given the same input). *See* Ex. 7 at 41 ("The pseudorandom sequences **appear to be generated by a truly random source** to anyone not knowing the method of generation."), 170 ("A pseudorandom bit generator (PRBG) is a deterministic algorithm which, given a truly random binary sequence of length k, outputs a binary sequence of length $l >> k$ which **'appears' to be random**."), 171 ("Two general

requirements are that the output sequences of a [pseudorandom bit generator] should be statistically **indistinguishable from truly random sequence** ….").

110. Pseudorandom number generators, especially those proven to be "cryptographically secure," generate values considered to be sufficiently "random" for practical purposes. The Handbook of Applied Cryptography discusses the use of cryptographically secure RNGs, including pseudorandom number generators. Ex. 7 at 170 ("**Cryptographically secure** pseudorandom bit generators are the topic of §5.5."), 399 ("Many protocols involving random numbers require the generation of **cryptographically secure** (i.e., unpredictable) random numbers. If pseudorandom number generators are used, an initial seed with sufficient entropy is required."). Technical dictionaries also confirm that random number generators would be understood to produce sufficiently random numbers for their intended purpose. *See* Ex. 9 at 172 (*random numbers*: "… in practice the **pseudorandom numbers generated by a particular program are sufficiently random for the purpose intended**"). A POSITA would have also understood that *true* RNGs can be impractical and would not reasonably be what the invention is limited to. Accordingly, a POSITA would consider pseudorandom number generators to be a type of random number generator.

111. Technical dictionaries also confirm that PRNGs generate values based on a uniform distribution where values are equally likely to be selected. Ex. 8 at 438 ("The process used in computers would be more properly called '**pseudorandom number generation**.'"); Ex. 9 at 172 (defining *random numbers* as "[n]umbers that are drawn from a set of permissible numbers and that have no detectable pattern or bias" and "have an **equal probability of being selected**. … [Computer] programs are designed to generate what are known as **pseudorandom numbers** … [that] are sufficiently random for the purpose intended.").

112. I understand that MARA argues that claim 7's use of a "*deterministic* function" for incrementing the output of claim 1's hash function indicates that that the "*random* number generator" in claim 1 cannot be deterministic. However, as noted above, the POSITA would understand "random number generator" to encompass true RNGs and deterministic RNGs. Using the word "deterministic" to describe the incrementing function in claim 7 does not change the ordinary meaning of "random number generator" in claim 1 (which encompasses deterministic RNGs).

113. I also understand that MARA argues that DSS supports a distinction between random number generators and pseudorandom number generators. I disagree. DSS does not say "random number generators" must exclude PRNGs. In fact, DSS indicates that both kinds are acceptable. Dkt. 52-17 at 2075. This confirms to the POSITA that "random number generator" in the '961 patent encompasses both types.

2. "seed"

114. I understand that Malikie proposes this term to mean "a value obtained from a random number generator that is used to as the starting value for a cryptographic key generation function." I further understand that the parties' dispute regarding "seed" is similar to random number generator, i.e., whether it encompasses pseudorandom values. The POSITA would understand that the ordinary meaning of "seed" encompasses pseudorandom values. That is, the ordinary meaning of seed is not limited to true random values (i.e., non-deterministic). Technical dictionaries confirm that the ordinary meaning of "seed" is not so limited. Ex. 8 at 471 (*seed*: "A starting value used in generating a sequence of random or pseudorandom numbers."); Ex. 11 at 436 (*seed*: "An initial number supplied to a computer's RANDOM NUMBER GENERATOR to begin a new number sequence.).

115. Moreover, the '961 Patent does not limit "seed" beyond its typical meaning to the POSITA. Each time it is used, it is used in the ordinary, non-limiting sense. '961 pat. at 2:40-43; 3:64-66; 4:37-39; 4:49-51; claims 1, 2, 9, 10, 16, 22-24. The patent's claims and specification consistently describe the "seed" as being generated from a "random number generator," which, as I explained above, is not limited to generators of true random values and encompasses generators of pseudorandom values. *See above.*

3. "The method of claim 1 wherein if said output is rejected, said output is incremented by a deterministic function and a hash function is performed on said incremented output to produce a new output; a determination being made as to whether said new output is acceptable as a key."

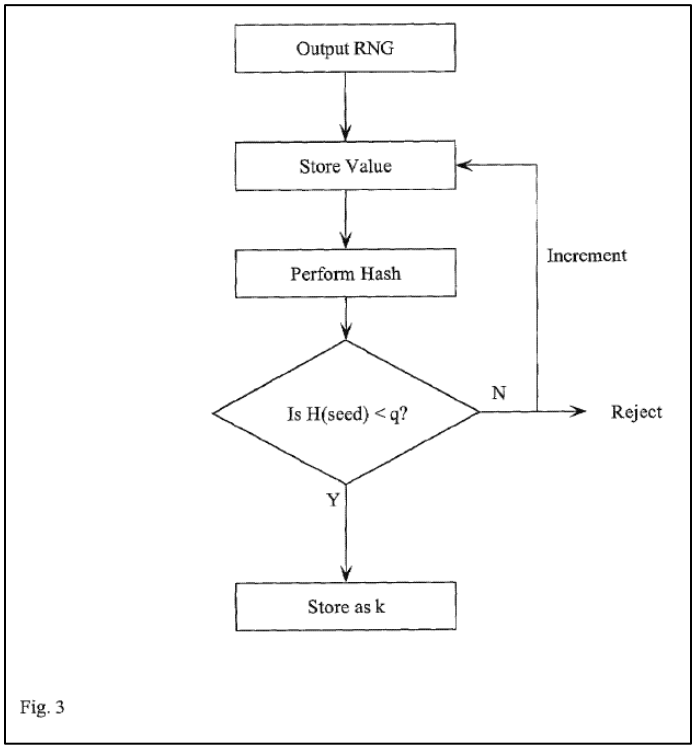
116. In my opinion, claim 7 is not indefinite to a POSITA. Rather, a POSITA would have readily understood claim 7 as written, in view of the intrinsic record.

117. Claim 7 depends from claim 1. Claims 1 and 7 recite, in relevant part:

> **1.** A method of generating a key k for use in a cryptographic function performed over a group of order q, said method including the steps of:
>
> [a] generating a seed value SV from a random number generator;
>
> [b] performing a hash function H( ) on said seed value SV to provide an output H(SV);
>
> [c – d] …
>
> [e] rejecting said output H(SV) as said key if said value is not less than said order q;
>
> [f] if said output H(SV) is rejected, repeating said method; and
>
> [g] …
>
> **7.** The method of claim 1 wherein if said output is rejected, said output is incremented by a deterministic function and a hash function is performed on said incremented output to produce a new output; a determination being made as to whether said new output is acceptable as a key.

118. As is clear from the claims themselves, claim 7 only occurs where the "output H(SV)" of Claim 1 is rejected. When this happens, element 1[f] requires that the method of claim 1 be repeated. Dependent claim 7 then adds additional limitations when the method is repeated.

The POSITA would understand that the specification teaches different ways of generating a key when an output is rejected. For example, "Upon rejection, the random number generator may generate a new value as disclosed in FIG. 2 or may increment the seed value as disclosed in FIG. 3." '961 Pat., 4:50-52. Claim 7 reflects the method of Fig. 3, which teaches the additional step of incrementing the rejected output by a deterministic function before hashing it. '961 Pat., Fig. 3; 4:18-31. A POSITA would therefore have understood claim 1 to encompass multiple ways of performing the same method, and that claim 7 merely claims another variant.



119. I understand that Dr. Koç opines that the specification does not provide the POSITA with reasonable certainty as to how claims 1 and 7 can be performed together. Dkt. 52-2 ¶90-93. I disagree. As explained above, the '961 patent teaches multiple ways to "repeat" the method when an output is rejected. One way is to have the random number generator generate another seed value, as required in claim 2. '961 Pat., Claim 2 ("wherein another seed value is generated by said random number generator"), 4:11-17. Another way involves additionally incrementing the

rejected output before hashing it, as required in claim 7. '961 Pat., Claim 7, 4:18-31. Accordingly, I disagree with Dr. Koç's opinion that claim 7 describes only "alternative" steps to claim 1, rather than additional requirements of claim 1's method.

I declare under penalty of perjury that to the best of my knowledge the foregoing is true and correct

Date: January 14, 2026

Dr. Paul Martin

Appendix A

harborexperts

Paul D. Martin, Ph.D.

Chief Scientist at Harbor Experts
Lecturer at Johns Hopkins University

P: 443.449.9006
E: paul@harborexperts.com

Profile

Dr. Martin is the Chief Scientist at Harbor Experts. He is also a lecturer at Johns Hopkins University, teaching courses on software development, computer and network security and cryptography. His research interests include embedded system security, operating system security, vulnerability analysis, reverse engineering, network protocol analysis, applied cryptography, and privacy-preserving protocols.

Education

2011-2016	*Ph.D. Computer Science, Johns Hopkins University, Baltimore, MD* *Securing Medical Devices and Protecting Patient Privacy in the Technological Age of Healthcare*
2011-2013	*M.S.E. Computer Science, Johns Hopkins University, Baltimore, MD*
2007-2011	*B.S. Computer Science, Johns Hopkins University, Baltimore, MD*

Industry Experience

2024-Present	*Harbor Experts*	**Chief Scientist**
2023-2024	*Harbor Labs*	**Vice President of Applied Research and Technology**
2023-Present	*Johns Hopkins University*	**Lecturer**
2018-2024	*Harbor Labs*	**Director of Firmware Security, Senior Research Scientist**
2013-2018	*Harbor Labs*	**Research Scientist**
2011-2016	*Johns Hopkins University*	**PhD Candidate, Research Assistant**
2013	*Applied Communication Sciences*	**Graduate Intern**
2011	*University of Michigan* *Inter-university Consortium for Political and Social Research* (ICPSR)	**Penetration Tester**
2009-2011	*Independent Security Evaluators*	**Security Intern**
2008-2010	*Johns Hopkins University* *Digital Research and Curation Center* (DRCC)	**Student Programmer**
2008	*Brandeis University*	**Freelance Programmer**

Teaching Experience

2025	*Intermediate Programming*	**Instructor**
2024	*Security and Privacy in Computing*	**Instructor**
2023	*Security and Privacy in Computing*	**Instructor**
2015	*Introduction to Hardware Hacking*	**Instructor**
2012-2014	*Security and Privacy*	**Teaching Assistant**
2011	*Practical Cryptographic Systems*	**Course Assistant**

harborexperts

Publications

P. Martin, D. Russel, M. Ben Salem, S. Checkoway, A. Rubin, Sentinel: Secure Mode Profiling and Enforcement for Embedded Systems, Proc. ACM/IEEE International Conference on Internet-of-Things Design and Implementation, (IoTDI '18).

P. Martin, M. Rushanan, T. Tantillo, C. Lehmann and A. Rubin, Applications of Secure Location Sensing in Healthcare. In the proceedings of ACM Conference of Bioinformatics, Computational Biology, and Health Informatics (BCB '16).

J. Carrigan, P. Martin, M. Rushanan, KBID: Kerberos Bracelet Identification. In the Proceedings of Financial Cryptography and Data Security (FC '16).

P. Martin, M. Rushanan, S. Checkoway, M. Green, A. Rubin. Classifying Network Protocol Implementation Versions: An OpenSSL Case Study. Technical Report 13-01, Johns Hopkins University (December 2013).

P. Martin, A. Rubin, and R. Bhatti, *Enforcing Minimum Necessary Access in Healthcare Through Integrated Audit and Access Control.* In ACM Conference on Bioinformatics, Computational Biology, and Biomedical Informatics Health Informatics Symposium (BCB-HIS), (September 2013)

Program Committees

IEEE Symposium on Security and Privacy (Winter 2024)
IEEE Symposium on Security and Privacy (Summer 2024)

Patents

System and Method for automatically extracting information from binary files for use in Database Queries	**US 10,762,214 B1**
System and method for network traffic profiling and visualization	**US 10,230,599 B2**
System and method for network traffic profiling and visualization	**US 9,667,521 B2**
System and method for network traffic profiling and visualization (Pending)	**WO 2015113036A1**
Healthcare privacy breach prevention through integrated audit and access control	**US 8,984,583 B2**
Healthcare privacy breach prevention through integrated audit and access control	**US 9,438,632 B2**

Current Research

Use of multimodal large language models for diagnosis of disease progression.
Use of large language models for certain types of technical research and analysis.
Automated binary version extraction for NVD cross-reference based on fuzzy matching.
Automated analysis of vulnerabilities in containers and virtual appliances.
Large-scale comparison of nature and kind of firmware vulnerabilities across and within product classes.

Expert Witness Engagements

CERTAIN FLASH-SPUN NONWOVEN MATERIALS AND PRODUCTS CONTAINING SAME

Case:	ITC Investigation No. 337-TA-1424
Description:	ITC Investigation related to nation-state threat actors, unfair competition and trade secret misappropriation.
Services:	Malware analysis and reverse engineering, cyber-attribution, computer forensics, reasonableness of security measures.
Expert Testimony at Deposition:	Washington, DC (September 6, 2025)

The Nielsen Company v. Hyphametrics, Inc.

harborexperts

Case:	Case # 23-136-GBW-CJB
Description:	Analysis related to audience measurement and identification systems and machine learning systems.
Services:	Source code review, expert report drafting, validity analysis.
Expert Testimony at Deposition:	Potomac, MD (February 5, 2025)
Expert Testimony at Trial:	Wimington, DE (July 24-25, 2025)

Jonathan Day and Michelle Dobek v. **Advanced Micro Devices, Inc**.

Case:	Case # 4:22-cv-00312-CRB
Description:	Litigation related to computer fTPM and TEE design characteristics.
Services:	Technical analysis and expert reports on security and performance characteristics of trusted execution environments and the design of NVRAM interfaces through SPI. Drafting of expert reports.
Expert Testimony at Deposition:	Washington, DC (February 7, 2025)

State of New Jersey v. **Paul Caneiro**

Case:	Indictment # 19-02-283; PG # 18-4915
Description:	Analysis related to reliability of specific types of computerized DNA analysis in criminal proceedings.
Services:	Source code review, expert report drafting.
Expert Testimony at Hearing:	Feehold, NJ (December 6, 2024)

Amazon.com, Inc. v. B.S.D. Crown, Ltd.,

Case:	IPR2025-00057
Description:	Analysis related to emulation and virtualization technology.
Services:	Invalidity analysis. IPR drafting.

MediaTek Inc. v. Daedalus Prime LLC

Case:	IPR2025-00100
Description:	Analysis related to security processors embedded in microcprocessors.
Services:	Invalidity analysis. IPR drafting.

McKinney v. **Corsair Gaming, Inc**.

Case:	Case # 4:22-cv-00312-CRB
Description:	Litigation related to computer memory performance.
Services:	Technical analysis and expert reports on characteristics of computer memory and forensic analysis of computer systems.

LoganTree v. Fossil

Case:	Case # 21-cv-0385-JDW
Description:	Litigation related to fitness tracking wearables.
Services:	Technical analysis and expert reports on validity and infringement.
Expert Testimony at Deposition:	Monrovia, MD (August 11, 2023)

Nielsen v. TVSquared

Case:	Case # 6:22-cv-00244-ADA
Description:	Litigation related to network-based audience measurement.
Services:	Technical analysis and expert reports on patentable subject matter.

ISI v. **Intuitive Surgical, Inc.**

Case:	Case # 3:21-cv-03496-VC
Description:	Litigation related to cryptographic security protections.
Services:	Technical analysis and expert reports on security and technical

harborexperts

aspects of surgical devices.
Expert Testimony at Deposition: Monrovia, MD (March 16, 2023)

Healthcare Advanced Risk Technologies, Inc. and Inspirien Holdings Corp. v. **Terrence Mills, Al.io. Corp., Jane Nemcova and VEUU, Inc.**
Case: Case # 2:22-cv-04192-JS-AYS
Description: Litigation related to timeline of software development and misappropriation of trade secrets.
Services: Source code and documentation review, source code copying analysis, expert report drafting.

Communication Technologies, Inc. v. Samsung Electronics America, Inc., and Samsung Electronics Co. Ltd.
Case: Case # 2:21-cv-444-JRG
Description: Litigation related to patents on secure erase processes.
Services: Source code review of algorithms related to secure erase processes, declaration on aspects of source code review process and representativeness of source code

Planck, LLC D/B/A Patch Media v. **Particle Media, Inc. D/B/A News Break, Et. Al.**
Case: Case # 20-cv-10959 (LGS)
Description: Litigation related to copyright.
Services: Review of web scraping facts, contracts and agreements and syndicated web feed technology, declaration on web scraping and syndicated web feed technology

WSOU Investments, LLC D/B/A Brazos Licensing and Development v. Cisco, Inc.
Case: Case # 6:21-cv-00128-ADA
Description: Litigation related to patents to patents on wireless network handoff, network management and authentication technologies.
Services: Source code review of accused products, declaration on aspects of source code review process.

United States **v. Laffon Ellis**
Case: Case # 2:19-cr-00369-DWA
Description: Analysis related to reliability of specific types of computerized DNA analysis in criminal proceedings.
Services: Source code review, expert report drafting.
Expert Testimony at Hearing: Monrovia, MD (December 20, 2021)

Sysmex Corporation and Sysmex America, Inc. v. **Beckman Coulter, Inc.**
Case: CA # 19-1642-RGA-CJB
Description: Litigation related to hematology analysis machine patents.
Services: Source code review. Expert report drafting.
Expert Testimony at Deposition: Monrovia, MD (November 22, 2021)

CERTAIN ROUTERS, ACCESS POINTS, CONTOLLERS, NETWORKS MANAGEMENT DEVICES, OTHER NETWORKING PRODUCTS, AND HARWARE AND SOFTWARE COMPONENTS THEREOF
Case: ITC Investigation No. 337-TA-1227
Description: ITC Investigation related to patents on wireless network handoff, network management and QoS technologies.
Services: Source code review, validity and prior art analysis, expert report drafting.
Expert Testimony at Deposition: Monrovia, MD (June 9-10, 2021)
Expert Testimony at Trial: Washington, DC (July 28, 2021)

Micro Focus, Inc. v. Insurance Services Organization

harborexperts

Case: DE Civil Action # 15-252-RGA
Description: Litigation related to unlicensed use of runtime environments, libraries and software compilers.
Services: Source code review. Binary reverse engineering and analysis, affidavit drafting, expert report drafting.
Expert Testimony at Deposition: Wilmington, DE (Feb 2, 2021)

loanDepot.com, LLC v. Sigma Infosolutions, Inc.
Case: AAA Case # 01-18-0001-5821
Description: Litigation related to software development practices.
Services: Source code analysis, experimentation, report drafting.
Expert Testimony at Deposition: Baltimore, MD (December 17, 2019)

Cypress Lake Software, Inc. v. Samsung Electronics America and Dell, Inc.
Case: Case # 6:18-cv-00030-RWS
Description: Litigation related to infringement of UX patents.
Services: Source code analysis, report drafting.
Expert Testimony at Deposition: Baltimore, MD (July 9, 2019)

Apple, Inc. Device Performance Litigation
Case: CA Civil Action # 18-md-02827-EJD
Description: Litigation related to business practices.
Services: Technical analysis and expert reports on security and technical aspects of mobile phone forensics.

Italian Antitrust Authority v. **Apple, Inc.**
Case: PS/11309
Description: Litigation related to business practices.
Services: Technical analysis and expert reports on security and technical aspects of software update processes.

Carl Zeiss AG and ASML Netherlands B.V. v. Nikon
Case: Case # 2:17-cv-07083-RGK (MRWx)
Description: Litigation related to patents on image detection algorithms.
Services: Source code review of algorithms related to image processing and detection algorithms, declaration on aspects of source code review process.

Decision Resources, LLC v. **Brigham Hyde, Precision Health Intelligence, LLC and Orr Inbar**
Case: MA Civil Action # 17-2834J
Description: Litigation related to timeline of software development and misappropriation of trade secrets.
Services: Source code and documentation review, development timeline analysis, affidavit drafting.

Litigation Support

Video Solutions v. Cisco, Inc
Case: Case # 2:23-cv-222-JRG
Description: Litigation related to videoconferencing and network performance optimization.
Services: Infringement analysis. Source code review.

OmniTracs, LLC v. Motive Technologies, Inc.
Case: Case # 3:23-cv-05261-RFL
Description: Litigation related to driver fleet management.
Services: infringement analysis. Source code review.

harborexperts

Robocast v. **Netflix, Inc**

Case:	Case # 1:22-cv-00305-JLH-CJB
Description:	Litigation related to hypermedia and multimedia node structures.
Services:	Invalidity and noninfringement analysis and report drafting. Source code review.

Proxense, LLC. v. Google, LLC

Case:	Case # 6:23-cv-00320-ADA
Description:	Litigation related to patents on biometrics and payment processing.
Services:	Claim construction analysis and report drafting.

Epic Games, Inc. & Anor vs. **Apple Inc & Anor**

Case:	Case # NSD 1236 of 2020
Description:	Litigation related to security and competition.
Services:	Document review, interviews, report drafting.

Softex LLC v. Absolute Software Corporation, Dell Technologies Inc, and HP Inc**.**

Case:	Case # 1:22-cv-01309-DAE
Description:	Litigation related to computer anti-theft technology.
Services:	Claim construction analysis, source code review, report review, binary reverse engineering and product testing.

Mediapointe, Inc v. Microsoft Corporation

Case:	Case # 2:22-cv-01009-MCS-MRW
Description:	Litigation related to content delivery networks.
Services:	Claim construction analysis, source code review, validity analysis, infringement analysis, report drafting.

Mediapointe, Inc v. Akamai Corporation

Case:	Case # 2:22-cv-06233-MCS-AFM
Description:	Litigation related to content delivery networks.
Services:	Claim construction analysis, source code review, validity analysis, infringement analysis, report drafting.

Lauri Valjakka v. **Netflix, Inc**

Case:	Case # 4:22-cv-01490-JST
Description:	Litigation related to content delivery networks.
Services:	Claim construction analysis, invalidity analysis, noninfringement analysis, report review.

Fanduel, Inc v. Winview, Inc

Case:	IPR 2022-01306, IPR-01307
Description:	Litigation related to patents on probability systems.
Services:	Document review, invalidity analysis, report review.

Proxense, LLC. v. Samsung Electronics America, Inc., and Samsung Electronics

Case:	Case # 6:21-cv-00210-ADA
Description:	Litigation related to patents on biometrics and payment processing.
Services:	Source code review, documentation review, product testing, validity analysis, infringement analysis, report drafting.

Global Eticket Exchange Ltd. vs. **TicketMaster LLC**

Case:	Case # 6:21-cv-00399-ADA
Description:	Litigation related to patents on electronic ticketing.
Services:	Source code review, documentation review, product testing, invalidity analysis, non-infringement analysis, report

harborexperts

drafting.

US Dominion, Inc. vs. Fox News Network
- Case: Case # N21C-03-257-EMD
- Description: Litigation related to defamation and voting machine security.
- Services: Source code review, documentation review, product testing, report drafting.

Centripetal Networks, Inc. vs. **Keysight Technologies, Inc.**
- Case: Case # 2:22-cv-0002-AWA-DEM
- Description: Litigation related to patents on network monitoring devices and security gateways.
- Services: Source code review, documentation review, product testing, noninfringement analysis, infringement analysis.

Milliman, Inc and Vigilytics LLC, vs. Gradient A.I. Corp.
- Case: Case # 1:21-cv-10865-NMG
- Description: Litigation related to breach of contract and source code copying.
- Services: Source code review, documentation review.

WSOU Investments, LLC D/B/A Brazos Licensing and Development v. Microsoft Corporation
- Case: Case # 1:18- 6:20-cv-00464-ADA, 6:20-cv-00460-ADA, 6:20-cv-00457-ADA,
- Description: Litigation related to patents on telephony management systems and skill-based matchmaking.
- Services: Source code review, documentation review, validity analysis, infringement analysis, report drafting.

10Tales Inc. v. TikTok PTE. Ltd.
- Case: Case # 1:18-cv-826-WCB
- Description: Litigation related to patents on user-adapted video streams.
- Services: Claim construction analysis.

Carriere v. **Symantec Corporation**
- Case: Case # 500-06-000894-176
- Description: Class action litigation related to product security.
- Services: Source code review, documentation review, report drafting.

IOENGINE, LLC v. Ingenico, Inc.
- Case: Case # 1:18-cv-826-WCB
- Description: Litigation related to patents on payment processing systems.
- Services: Source code review, documentation review, validity analysis, infringement analysis, report drafting.

IOENGINE, LLC v. PayPal Holdings, Inc.
- Case: Case # 1:18-cv-452-WCB
- Description: Litigation related to patents on payment processing systems.
- Services: Source code review, documentation review, validity analysis, infringement analysis, report drafting.

AGIS Software Development LLC v. **Uber Technologies**
- Case: Case # 2:21-cv-00026-JRG-RSP
- Description: Litigation related to patents on map overlays and messaging systems.
- Services: Source code review.

Finjan v. **Palo Alto Networks**
- Case: Case # 4:14-CV-04908-PJH

harborexperts

Description:	Litigation related to patents on malware scanning gateways.
Services:	Invalidity analysis, Claim construction analysis, source code review.

Huawei Technologies Co. v. **Verizon Communications Inc.**

Case:	Case # 6:20-CV-00090
Description:	Litigation related to patents on malware scanning gateways with cloud components.
Services:	Source code review, non-infringement analysis.

Epic Games, Inc. vs. **Apple, Inc.**

Case:	Case # 4:20-cv-05640-YGR-TSH
Description:	Litigation related to security and competition.
Services:	Document review, interviews, report drafting.

Philips North America LLC ; Koninklujke Philips N.V. vs. **Summit Imaging Inc.**

Case:	Case # 2:19-cv-01745-JLR
Description:	Litigation related to third-party repair services.
Services:	Source code review, document review, report drafting.

California Physicians Service, Inc D/B/A Blue Shield of California vs. Healthplan Services Inc,

Case:	Case # 3:18-cv-3730
Description:	Litigation related to software development practices and breech of contract.
Services:	Document review, source code review, report drafting.

Finjan v. **Qualys**

Case:	Case # 4:18-cv-07229-YGR
Description:	Litigation related to patents on vulnerability assessment products.
Services:	Invalidity analysis, Non-infringement analysis, source code review, report drafting.

Finjan v. **Sonicwall**

Case:	Case # 5:17-cv-04467-BLF-HRL
Description:	Litigation related to patents on malware scanning gateways.
Services:	Invalidity analysis, Non-infringement analysis, source code review, report drafting.

TecSec Inc. v. Cisco and Oracle

Case:	Case # 1:10-cv-115 LO-TCB
Description:	Litigation related to patents on hardware accelerated cryptographic processors.
Services:	Infringement analysis, validity analysis, source code review, report drafting.

Blackberry Limited v. Facebook, Inc.

Case:	Case # 2:18-cv-01844-KSx
Description:	Litigation related to patents on agent-based network monitoring, configuration and security systems.
Services:	Infringement analysis, validity analysis, source code review, report drafting.

Uniloc, Inc. v. **Big Fish Games, Inc.**

Case:	Case # 2:16-cv-00741-JRG
Description:	Litigation related to patents on hardware cryptographic chips.
Services:	Non-infringement analysis, report drafting, source code review.

SPEX Technologies, Inc. v. **Toshiba America Electronic Components, Inc., et al.**

harborexperts

Case:	Case # 8:16-cv-01800-JVS
Description:	Litigation related to patents on hardware cryptographic chips.
Services:	Non-infringement analysis, report drafting.

Symantec Corporation v. Zscaler, Inc.

Case:	Case # 3:17-cv-04414-JST,
Description:	Litigation related to patents on security gateways, URL filtering and categorization
Services:	Infringement analysis, assignor estoppel, document review.

Koninklijke Philips v. **Microsoft Inc.**

Case:	Case # 4:18-cv-01885-HSG,
Description:	Litigation related to patents on secure cryptographic protocols
Services:	Non-infringement analysis, validity analysis, claim construction analysis, document review, source code review.

Netfuel, Inc. v. Cisco Systems, Inc.

Case:	Case # 5:18-cv-2352-EJD
Description:	Litigation related to patents on agent-based network monitoring, configuration and security systems.
Services:	Infringement analysis, claim construction analysis, source code review, report drafting.

Byrd et al. v. Aaron's, Inc., et al.

Case:	PA Civil Action # 1:11-cv-00101-SJM-SPB
Description:	Class action litigation related to privacy.
Services:	Attend depositions, source code review, report drafting.

Finjan v. **Juniper Networks**

Case:	Case # 3:17-cv-05659-WHA
Description:	Litigation related to patents on malware scanning gateways.
Services:	Invalidity analysis, non-infringement analysis, source code review.

Grace et al. v. **Apple Inc.**

Case:	Case # 5:17-cv-00551-LHK (NC)
Description:	Litigation related to device performance and service outages.
Services:	Mobile forensics and device analysis, report drafting, source code review, document review, technical analysis and argument construction.

Rimini Street, Inc. v. Oracle International Corporation, et al.

Case:	Case # 2:14-cv-01699 LRH-CWH
Description:	Litigation related to false claims on security.
Services:	Large-scale testing of IPS techniques for including custom test infrastructure and implementation of techniques to block exploitation of vulnerabilities, technical analysis of vulnerabilities.

Finjan v. **Symantec Corporation**

Case:	Case # 14-cv-02998-HSG
Description:	Litigation related to patents on malware scanning gateways, endpoint protection and firewalls.
Services:	Build and/or test software for Windows, invalidity argument strategy, non-infringement argument strategy, report preparation, source code reviews.

Strikeforce, Inc. v. **Entrust et al.**

Case:	Case # 1:17-cv-00309

harborexperts

Description:	Litigation related to patents on authentication technologies.
Services:	Invalidity argument development, non-infringement argument development, report drafting.

Sony Corporation, Inc. v. Arris

Case:	Inv. # 337-TA-1049
Description:	Litigation related to patents on television streaming devices and/or services.
Services:	Validity argument development, source code review of entire platform codebase including numerous embedded platforms, infringement argument development.

Kudelski SA, Nagra USA, Inc., Nagravision SA, and OpenTV, Inc. v. Comcast Corporation

Case:	Case # 2:16-cv-1362-JRG, Inv. # 337-TA-1049
Description:	Litigation related to patents on television streaming devices and/or services.
Services:	Validity argument development, source code review of entire platform codebase including numerous embedded platforms, infringement argument development.

Amazon.com Inc., Hulu, LLC, and Netflix, Inc. v. Uniloc Luxembourg S.A.

Case:	IPR 2017-00948
Description:	Litigation related to patents on DRM protection for content distribution
Services:	Prior art search, PGR Preparation, IPR preparation.

PhishMe v. **Wombat Technologies, Inc.**

Case:	Case # 16-403-LPS-CJB
Description:	Litigation related to patents on anti-phishing training technologies.
Services:	Prior art search, PGR Preparation, IPR preparation.

Nader Asghari-Kamrani and Kamran Asghari-Kamrani v. **United States Automobile Association**

Case:	Case # 2:15-cv-478
Description:	Litigation related to patents on authentication technologies.
Services:	Prior art search, invalidity argument strategy, non-infringement argument strategy, source code reviews

Vir2us v. **Invincea Inc. and Invincea Labs, LLC**

Case:	Case # Case 2:15-cv-00162-HCM-LRL
Description:	Litigation related to patents on virtualization and automated corruption repair.
Services:	Prior art search, invalidity argument strategy, non-infringement argument strategy, source code reviews

Palo Alto Networks v. Finjan

Case:	IPR 2016-00159, IPR 2016-00151, IPR 2015-01974, IPR 2015-02001, IPR 2015-01979
Description:	Litigation related to patents on malware scanning gateways and firewalls,
Services:	Prior art search, patent interpretation, IPR preparation support, claim chart review

TVIIM v. **McAfee**

Case:	Case # 3:13-cv-04545-VC
Description:	Litigation related to patents on vulnerability scanning
Services:	Build and test software for SPARC/Linux/Windows, patch out license checks/crack software (with permission), obtain hard-to-find legacy software, prior art and non-

harborexperts

infringement argument strategy support, source code reviews, prior art search

Al Cioffi et al. v. Google

Case:	Case # 2:13-cv-103-JRG-RSP
Description:	Litigation related to patents on browser sandboxing and process isolation.
Services:	Code review/software testing to collect evidence of infringement, Infringement argument preparation support, claim chart review

Rovi Solutions & Veracode v. Appthority

Case:	Case # 12-10487-DPW
Description:	Litigation related to patents on static debugging tools
Services:	Source code review refuting opposing expert testimony

Analysis, Design and Development Clients

Arai	ICU Medical
Baxter	Inexto
Bigfoot Medical	Intuitive Surgical
BT Group	Merlin
Cardiac Sciences	Orpheus
Dyadic	Security First Corporation
Fresenius	Texas Instruments
HLFIP Holdings	Thesys
Hospira	Vaxxin

Pre-Harbor Labs Security Design and Software Development Experience

2013 At *Applied Communication Sciences*

Role:	Graduate Intern
Technologies:	JavaScript, Python, Tcpdump, Wireshark

- Developed extensible real-time traffic visualization tool to chart and analyze high-volume tcpdump streams of lossy metropolitan-area mesh network traffic.

2011 At *University of Michigan ICPSR*

Role:	Penetration Tester
Technologies:	Numerous Security Tools, Amazon EC2, VMware VSphere

- Conducted wide-scale penetration testing on virtualized cloud-based systems meant to be secure environments for researchers to store confidential results
- Created formal threat model document detailing potential security vulnerabilities from all possible attack vectors
- Wrote two reports detailing results from penetration test

2009-2011 At *Independent Security Evaluators*

Role:	Security Intern
Technologies:	C++, C#, Dalvik Bytecode, Gcov, GDB, Javascript, Peach Fuzzer, Python, RegEx, XML, Wireshark

- Created log parsing framework to analyze 20+ log file formats
- Assisted with malware testing, research and analysis
- Reverse engineered DRM schemes in Android and IOS applications
- Researched and prototyped secure cryptographic mail delivery system

harborexperts

- Developed web crawler to collect file sets for use in fuzzing
- Wrote code-coverage analysis tool for constructing minimum file set for fuzz testing
- Wrote fuzzing plugins using Peach Fuzzer framework and reverse engineered binary file specifications
- Wrote Internet Explorer and Chrome extensions for cryptographic proxy system
- Created and debugged network protocols for use in network protocol testing
- Wrote and debugged unit tests in C++ and Python for proprietary disk-encryption system

2008-2010 At *Johns Hopkins University DRCC*
Role: Student Programmer
Technologies: DOM, Java EE, JSP Perl, SAX, XSLT

- Drafted a report detailing security recommendations for an NSF funded data conservancy project
- Set up and deployed a Fedora digital repository with the Islandora frontend
- Ported IRStats statistics package to the DSpace information repository XMLUI
- Wrote batch importer that is now used to import more than 20 digitized books a week into DSPace repository

2008 At *Brandeis University Information Technology Services Hardware Repair Shop*
Role: Freelance Programming Consultant
Technologies: Java, Visual C++, VBScript

- Sole programmer on project to interface Request Tracker ticketing system with Brother PT- 9500PC Label Printer

Technical Skills

Languages	BASH, C, C++, C#, HTML, Java, JavaScript, Objective-C, Python, Perl, PHP, Regular Expressions, SQL, XML
Architectures	6502, 8051, 8080, ARM Cortex-M, ARMv7, ARMv8, AVR, m68k, MIPS, MSP430, PIC, SPARC, PowerPC, x86, x86-64, Z80
Operating Systems	Android, ChromeOS, FreeBSD, iOS, OpenBSD, Linux, macOS, Windows
DevOps and Development Tools	Ansible, Ant, BitBucket, Confluence, Docker, gdb, git, GitHub, GitLab, Gradle, Hadoop, jad, jd-gui, Jira, Maven, make, MySQL, PostgreSQL, subversion, Trello, Vagrant, valgrind
Security Tools	Aircrack-ng, apktool, binwalk, bulk-extractor, Burp suite, Charles Proxy, curl, dex2jar, ftk, hashcat, IDA Pro, Metasploit, mitmproxy, Nessus, nmap, OpenSSL, ophcrack, p0f, Scalpel, skipfish, snort, sslstrip, sslyze, Volatility, Web Scarab, wget, Wireshark
Cloud and Virtualization	AWS, Azure, Bhyve, KVM, LXD, QEMU, virt-manager, VMware, Xhyve

Honors, Societies and Awards

- Member Upsilon Pi Epsilon International Computer Science Honor Society
- Member Institute for Electrical and Electronics Engineers (#97507890)
- Member Association for Computing Machinery (#9700346)

At Johns Hopkins University

- Computer Science Department Outstanding Teaching Assistant Award (2014)

harborexperts

- Treasurer, Upsilon Pi Epsilon International Computer Science Honor Society (JHU Chapter)
- Computer Science Department Faculty Liaison Czar
- Student Representative to the Computer Science Undergraduate Curriculum Planning Curriculum Committee

Certifications

- Certified Six Sigma Black Belt